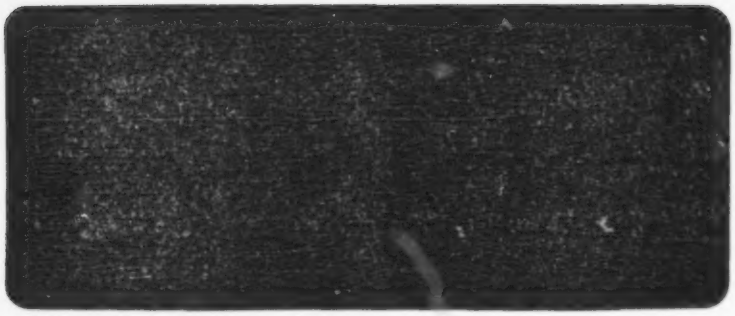
98-629



*intelligent*


**Applied Silicon Inc. Canada**

220 - 2427 Holly Lane, Ottawa, Ontario K1V 7P2 • Tel: (613) 738-2434 • Fax: (613) 738-0750

105-03416/

CONTRACT NUMBER W7714-5-9868

# FINAL REPORT
# OF THE SILO BEACON SOURCE

Contract Number W7714-5-9868

Revision 1.0

prepared for: Department of National Defence
Defence Research Establishment Ottawa
3701 Carling Avenue
Ottawa, Ontario
K1A 0K2

Scientific Authority: Denis Lamothe
(613) 998-2040

by
John Curnow

**APPLIED SILICON INC. CANADA**
220-2427 HOLLY LANE
OTTAWA, ONTARIO, CANADA
K1V 7P2
(613) 738-2434

30 May 1996

Document #974-55126-12

## SUMMARY

This document is the final report for the remote controlled frequency agile silo beacon RF source (SBS). It presents the test results after the construction of the SBS and the changes made since the design specification document was written. The 'silo' is located near Luskville, Quebec, and is 13.5 kilometers away from the Experimental Array Radar System (EARS) at the Area Five radar test site. One of two Portable Interference Frequency Sources (PIFS) was dismantled to supply the parts for the silo source. PIFS 1 is still intact and operational. This task is covered under section 2.9 of the amended Statement of Work for the W7714-5-9868 contract.

The silo beacon source has the same functionality as the PIFS, including; RF output from 8.9 to 9.4 gigahertz, CW, noise or doppler modulation, doppler frequency from one to 7813 hertz. The following functions are remotely controlled via a spread spectrum RF modem link; RF on/off, RF frequency, modulation selection, doppler frequency, RF transmit signal amplitude. A personal computer at the Area Five site runs a control program to communicate with and control the silo source. The source can also be run in local testing mode using a portable terminal.

The silo source contains a small single board computer (SBC) and a digital input/output (I/O) board. The SBC receives, interprets and implements either the remote or local commands. Connections are made from the I/O board to each controlled component. A control program, written in C++, runs on the SBC. A series of command codes and messages are only one byte long, except for the code to set the doppler frequency, which is two bytes long.

Several key design modifications were implemented to convert the PIFS hardware for use as the SBS. A 24 VDC power supply was added to replace the two gell-cell batteries that powered the PIFS. The 'standby' and low voltage shutdown modes were eliminated. Two AC fans and a heater, all under thermostatic control have been added. The resident and control software was modified slightly. A yagi antenna for the modem link has been mounted on the SBS enclosure. The new SBS enclosure was made very weather resistant, with ventilation filters and no exposed switches or connectors. It is much larger than the PIFS enclosure but is the same type of box.

## TABLE OF CONTENTS


| | | |
|---|---|---|
| 1.0 | INTRODUCTION | 1 |
| 2.0 | CONSTRUCTION | 2 |
| 3.0 | TESTING | 5 |
| | 3.1 RF/IF Testing | 5 |
| | 3.1.1 RF/IF Level Measurements | 7 |
| | 3.1.2 Digital Attenuator Measurements | 7 |
| | 3.2 Cooling Test | 8 |
| | 3.3 Heating Test | 9 |
| 4.0 | DESIGN CHANGES | 10 |
| 5.0 | INSTALLATION ON THE SILO | 11 |
| 6.0 | MAINTENANCE | 12 |
| 7.0 | OPERATION | 12 |

*APPENDIX A* - **COMPLETE PARTS LIST**

*APPENDIX B* - **RESIDENT CONTROL SOFTWARE SOURCE LISTING**

REFERENCES

ABBREVIATIONS AND ACRONYMS

## LIST OF TABLES


| *TABLE NUMBER* | *TITLE* | *PAGE* |
|---|---|---|
| 1 | SBS RF and IF Component Levels Measurements | 6 |
| 2 | Ten Megahertz Reference Oscillator Levels | 7 |
| 3 | Digital Attenuator Amplitude Measurements | 8 |
| 4 | Monitor Connector Details | 10 |

## 1.0 INTRODUCTION

This document is the Final Report for the "Design Upgrade and Hardware Modifications to the Radio Frequency Silo Beacon Source" task. This task is covered under section 2.9 of the amended Statement of Work for the W7714-5-9868 contract.

A Hardware Design Specification Document (HDSD) was written earlier, in May 1996. It covered the hardware and software design of the SBS upgrade in detail. The HDSD was written after most of the design had been executed, for both the hardware and software. Therefore, it is an accurate design document, with minor changes and exceptions. These changes and exceptions are detailed in this final report. Since so few changes have been made since the HDSD was written, the information contained in it will not be repeated in this document. For a complete final detailed design the reader should read the HDSD before reading this final report.

The month of May 1996 was the only time allowed to complete this task. Although the work was completed on time, some sacrifices were made in the design due to the time frame. Some aspects of the PIFS design which would otherwise have been improved could not be because of the limited time. This includes a re-design of the power supply section, elimination of the standby and under voltage circuits, improvements to the doppler generation circuit and some re-wiring. The enclosure was chosen because of its availability, not because it was the best one for the application. Remote control of the parabolic antenna aim might have been possible if more time and equipment money was available.

The basic design and functionality of the PIFS was proven during recent field trials. This design has been largely carried over to the SBS. PIFS 2 was dismantled to supply the components for the SBS. The main design modifications included a new enclosure, upgraded cooling, a new heating system, elimination of the under voltage and standby functions and inclusion of a 24 VDC power supply to replace the batteries. The resident control software was modified to reflect the hardware changes. The remote control computer software was also modified and is able to control PIFS 1 and the SBS.

## 2.0 CONSTRUCTION

Before PIFS 2 was dismantled a quick functional test was done. It worked well, although time did not allow in-depth testing. Each PIFS had identical Noise Com noise sources. One of these sources failed, so it was decided to put the one working source in the SBS. There are no noise source backups. One of the CTI RF sources also had a problem. It could not be switched to 8.9 GHz with a frequency step greater than 50 MHz. An attempt to do so would produce an incorrect frequency. This synthesizer was originally in PIFS 2 but was put into PIFS 1 so the fully functional synthesizer could be put into the SBS.

Two main criteria drove the design of the SBS. The first was the remoteness of the location. The 'silo' is located near Luskville, Quebec, and is 13.5 kilometers away from the Experimental Array Radar System (EARS) at the Area Five radar test site. It takes about 45 minutes to travel to the silo. The SBS is mounted high up on the silo and must be lowered to the bottom of its travel to be accessible. Even then it cannot be accessed without the use of a ladder. This makes servicing difficult and time consuming. Reliability of the SBS is important due to the difficulty of servicing. It must operate reliably over the long term, in all types of year-round weather.

The second criteria driving the design was the time frame. Only readily available, off-the-shelf parts could be used, such as the SBS enclosure and power supply. Limited funds severely restricted what could be purchased. It was not possible to have any mechanical work done by the CRC Model Shop since they could not even start a work order until near the end of May.

Mr. Mike Boyle of the CRC Shop was used for relatively simple and quick mechanical work. This included the interior mounting plate, the lower skirt and fan cutouts in the enclosure. Formal work orders were not required to obtain his assistance. Verbal requests were sufficient and he did the work when he was available. Fortunately he was able to do the requested work within a day of each request.

One of the first design decisions was the choice of the enclosure. The steel Hammond box that was used was not the first choice but the preferred enclosures were not available in time.

The Hammond model 1418 E4 N8 enclosure is about the largest that could physically fit in the space where the previous beacon source box was located. It was hoped that a slightly smaller and lighter enclosure without a hinged lid could be obtained but this was not possible. This enclosure is basically just a larger version of the PIFS enclosure. They are both from the Hammond 1418 enclosure series. The SBS enclosure is 24 inches by 20 inches by 8 inches deep.

After the enclosure was purchased, the interior layout was designed. The steel mounting plate included with the 1418 box was replaced with a larger and thicker aluminum plate. This was done so that holes could easily be drilled and tapped into the mounting plate. One of the mechanical design requirements was that all the major components could be removed without removing the mounting plate. This could only be done if the plate could be tapped, so nuts were not required under the plate. Only one component uses nuts underneath the plate. The noise source is mounted this way because it was too difficult to reliably line up the mounting holes. It is possible to remove the noise source without removing the mounting plate since it is mounted right at the edge of the plate. The nuts can be accessed by reaching underneath the plate, between the plate and the side of the enclosure.

After the enclosure was received, Mike Boyle made the mounting plate and cut the openings for the two fans. He also made the interior center baffle plate. All the other openings in the enclosure, baffle plate and mounting plate were done at Area Five using drills or borrowed punches. Three 'L' brackets were made to hold the baffle plate to the mounting plate. The baffle plate has two purposes. It is used to mount the computer boards on their standoffs and the voltage regulators on their heat sinks. It is also used to direct the airflow through the case so that fan-forced air goes over all the heat producing components. All the RF components are mounted on one side of the interior plate. Most of the power supply components are mounted on the other side of the center plate. The air flow goes over the RF components first, giving them the maximum cooling. The mounting plate also acts as a heat sink to dissipate the RF component heat.

It is not possible to put the mounting plate into the enclosure with all the components mounted on it. Since it is bigger than the original steel plate, it is a tighter fit and is

more difficult to get in or out of the enclosure. It can only go in if the fans, +24 volt power supply, EMI filter, quadrature board, center plate, power supply board, daughter board and RF modem are not in place. All the other components can be mounted on the mounting plate. As much mechanical and electrical work as possible was done with the mounting plate outside of the enclosure. This included drilling and tapping all the mounting holes. The holes and cutouts in the enclosure were made while the mounting plate was being prepared. The RF modem and quadrature board were test mounted on standoffs.

After all the holes were drilled and tapped in the mounting plate the RF components were mounted on it. Each RF component had been previously separately bench tested. Heat sink compound was used under the noise source, synthesizer, 10 MHz source and the RF power amplifier. The semi-rigid 0.141 inch coaxial cables were then made. The DC power supply wires to each component were made and assembled onto the plate. Three cutouts are made in the center plate to allow access for the wires. Two are at the bottom and one is at the top.

After test fitting most of the enclosure mounted devices, all the devices were removed from it, including the carrying handles. The enclosure was spray painted white at Area Five. If time had permitted, it would have been sprayed in the CRC Model Shop. The colour white was chosen to minimize the solar heating effects. After the paint dried the components were put back in the enclosure.

The lower metal skirt was designed and made while the mounting plate was being worked on. Mike Boyle made the four sides of the skirt. The skirt is made of 0.040 inch thick aluminum plate. It is 58 centimeters long by 7.7 cm deep by 15.2 cm wide. The two short end pieces extend past the long side pieces by 7.7 cm to give more protection from precipitation. 1.3 cm aluminum angle pieces are screwed just over the lower lip of the long skirt side pieces. This forms two millimetre wide channels into which the outer screen is slid and held. The skirt was test assembled onto the enclosure and then removed. It was not put back on until everything else was mounted. It has not been painted.

## 3.0 TESTING

Testing was done in stages, working from single components up to the completed SBS on the silo. Almost all the components had been used in PIFS 1 or PIFS 2. Each RF component was individually bench tested before it was mounted in the SBS. Unfortunately the HP8566 spectrum analyzer was not available so the old HP141 was used instead. It does not give an accurate frequency reading and is more difficult to use. It is not possible to get hard copy screen printouts because the 141 does not have an HPIB interface.

### 3.1 RF/IF Testing

All the individual components worked well on the bench. Each semi-rigid coaxial cable was carefully made and tested, with good results. Further RF/IF testing could only be done after all the RF/IF components were connected and functioning inside the SBS.

The DC power to all the components was disconnected and the AC power was applied once everything but the fans was installed. The DC voltages were checked. The component power lines were then connected and the SBS was powered up. Control was exercised in local mode only. There was a small bug in the display format in local mode which was fixed by disabling the key echo on the hand-held terminal.

A horn was put on the RF output and a similar horn was used on the input to the HP141 spectrum analyzer. They were positioned about six meters apart. The SBS RF signal was displayed on the 141. All the functions of the SBS were then checked, under local control. It was difficult to check the doppler frequency with the 141 due to its instability at a high resolution but it could be seen that a doppler shift had occurred at approximately the correct frequency. The doppler and noise modulate the upper sideband, therefore the transmit frequency shifts upward when a doppler frequency is used.

Remote control was then tested. The modified PIFSCON.C control program for the PC was used with the new SBS.C program running on the SBS computer. All the remote functions worked well. Some minor bugs were fixed in the PIFSCON program such as not always displaying the status when switching control from one unit to the other.

| COMPONENT | INPUT(dBm) | LOSS /GAIN (dB) | OUTPUT (dBm) |
|---|---|---|---|
| SYNTH. SOR2 | | | 14.12 |
| Attenuator, AT2 | 14.12 | -6.00 | 8.12 |
| Cable | 8.12 | -0.24 | 7.88 |
| Attenuator, AT3 | 7.88 | -5.83 | 2.05 |
| Attenuator, AT4 | 2.05 | -2.88 | -0.83 |
| SSBM, SSB1 | -0.83 | -10.62 | -11.45 |
| Cable | -11.45 | -0.40 | -11.85 |
| Isolator, AT6 | -11.85 | -0.33 | -12.18 |
| Attenuator, DAT1 | -12.18 | -1.96 | -14.14 |
| Cable | -14.14 | -0.2 | -14.34 |
| RF SWITCH, SW1 | -14.34 | -1.05 | -15.39 |
| AMP, AMP1; +19 dBm O/P | -15.39 | 14.33 | -1.06 |
| Cable | -1.06 | -0.27 | -1.33 |
| Isolator, AT7 | -1.33 | -0.60 | -1.93 |
| AMP, AMP2; +29 dBm O/P | -1.93 | 30.97 | 29.04 |
| Cable To J1 | 29.04 | -0.30 | 28.74 |
| Cable to ANTENNA | 28.74 | -2.24 | 26.50 |

Table 1
**SBS RF and IF Component Levels Measurements**

### 3.1.1 RF/IF Levels Measurements

The RF and IF levels through each component were checked after the SBS had been on for several hours. All the measurements were made with the RF set to the center of the band, at 9.150 GHz. Table 1 shows all the measurements. No modulation was used and the attenuator was set to zero. The level at the SMA RF output connector is +28.74 dBm. The loss through the coaxial cables to the antenna is 2.24 dB, yielding an input level to the parabolic antenna of +26.5 dBm.

Level measurements were made at the output of the 10 MHz reference oscillator and at the output of the RG174 cable connecting it to the synthesizer. These measurements are shown in table 2.

| COMPONENT | INPUT (dBm) | LOSS/GAIN (dB) | OUTPUT (dBm) |
|---|---|---|---|
| 10 MHz Oscillator, SOR1 | | | 7.04 |
| RG 174 CABLE | 7.04 | -0.03 | 7.01 |
| 6 dB Attenuator, AT1 | 7.01 | -6.00 | 1.01 |

Table 2
**Ten Megahertz Reference Oscillator Levels**

### 3.1.2 Digital Attenuator Measurements

The attenuator was checked after the SBS was assembled and working in T-80. The HP436A power meter was put on the RF output to make accurate amplitude measurements. All the attenuation was removed and the meter was set to relative mode. Table three shows the measurement results. Each of the individual sections was checked first. It can be seen that each section is lower than it should be. The maximum specified error is ±1 dB. It is definitely outside the specification but it is adequate for the SBS application.

| Setting (dB) | Actual Attenuation (dB) |
|---|---|
| 1 | 0.43 |
| 2 | 1.11 |
| 4 | 2.86 |
| 8 | 7.06 |
| 16 | 15.16 |
| 32 | 30.38 |
| 7 | 5.93 |
| 10 | 9.14 |
| 15 | 14.06 |
| 20 | 18.76 |
| 30 | 28.87 |
| 40 | 38.40 |
| 50 | 47.38 |

Table 3
**Digital Attenuator Amplitude Measurements**

3.2 Cooling Test

The two fans were installed after all the other components were in place. The inlet fan just touches one of the connectors on the SBC, but this is not a problem. The thermostat was shorted and power was applied. The airflow through the enclosure was qualitatively checked with various combinations of filters. It was decided to use the full three layer filters on the outside of each fan. They use the light green open cell foam sandwiched between corrugated layers of metal open weave screen material. Although these filters restrict the airflow, it is still adequate for cooling with the two fans operating. If time allowed, a temperature monitor circuit could have been installed which would trip an alarm bit if the internal temperature was too hot or too cold.

The temperature probe was taped to the mounting plate beside the 120°F thermostat. The SBS was powered on with the lid closed. The temperature gradually rose over a period of about 45 minutes. At 49.6°C the fans came on. The temperature gradually cooled off until the fans turned off at 32.8°C. This took about fifteen minutes. The temperature gradually rose again. This fan on/off cycle continued. It is estimated that on a very hot sunny summer day the fans will stay on continually. Under slightly cooler ambient conditions the fans should cycle on and off. When the outdoor temperature is below about 25°C the fans should cycle. During average winter days the fans will never come on.

### 3.3 Heating Test

The new heating system using the power resistors and 35 degree Fahrenheit thermostat required testing. This was done after the entire heating circuit was assembled onto the plate. The only facility readily available for cooling the plate was the small environmental chamber at Area Five. It was an old Associated Testing Labs model SLHU-1-CRLC environmental chamber. The chamber was far too small to accommodate the entire plate, however, the plate would just fit in diagonally with about a third of it protruding out the front of the chamber.

A piece of rigid blue styrofoam insulation was cut to cover the opening below the plate. Another was cut to cover the opening above the plate. Tape and styrofoam packing chips were used to cover the opening that remained. A temperature probe was taped to the plate beside the thermostats, with its digital display on top of the chamber. A multimeter was used to monitor the AC voltage across one of the power resistors.

The chamber was set to refrigerate to a temperature of about -20 degrees Celsius. The resistor voltage was monitored as the temperature dropped. When the temperature reached -8.5°C one of the thermostats popped on. The chamber was then turned off. The temperature gradually rose and the voltage turned off at +8.3°C. A slower rate of temperature change was then tried.

The chamber was set to cool to -10°C. At -5.9°C one of the thermostats turned on. At -8.9°C the other thermostat turned on. The chamber was left on refrigerate for a few more minutes and then turned off. As the temperature increased one of the thermostats

turned off at +1.4°C and the other turned off at +10.4°C. Therefore, the heaters were turned on between -5.9°C and +10.4°C. This short test proves the functioning of the thermostats and shows the effectiveness of the heater elements. Under actual outdoor conditions, the heating or cooling would be much more gradual, therefore, the heater temperature span would probably be narrower. It is estimated that a very gradual cool down would turn the heaters on at a warmer temperature. Conversely, as the heaters gradually warm the interior, they should turn off at a lower temperature than +10°C.

## 4.0 DESIGN CHANGES

Very little has changed since the design document was written. Printed labels have been affixed to the outside of the enclosure for the connectors, fuse and switch. It is not known if the labels will last outdoors but time did not permit more permanent labelling. As indicated in section 3.2, the filters on both fans are three layer, with foam sandwiched between layers of screen.

The connector used on the underside of the enclosure for monitoring is a Bendix PT02E-12-10S. It is mounted beside the RF output SMA connector. Table four details the pinout of this connector.

| Connector Pin # | Signal | Internal Wire Colour | Destination |
|---|---|---|---|
| A | ground | black | terminal block |
| B | +24 VDC | red | terminal block |
| C | +15 VDC | yellow | terminal block |
| D | CTI synth. alarm | white/yellow | I/O board J1-28 |
| E | +5 VDC | blue | terminal block |
| F | -5 VDC | violet | power supply board J1-9 |
| G | -15 VDC | white | terminal block |
| H | +6.6 VDC | white/green | daughter board J1-6 |

Table 4
**Monitor Connector Details**

All the DC voltage signals except the +6.6 and -5 VDC are taken from the terminal strip. Spade lugs are used on the ends of the wires to the terminal strip. The +6.6 VDC components are on

the power supply daughter board. A jumper wire was soldered from the pin 2 connector post to the pin 6 connector post. Pin 6 was unused but is now the +6.6 VDC monitor output, therefore, the white wire with the green tracer goes from the monitor connector pin H to the daughter board pin 6.

The J1 pin 9 connection on the main power supply board is used to access the -5 VDC for monitoring. This voltage is only used internally on this board. Therefore, the violet wire goes from the power supply board J1-9 pin to the monitor connector F pin.

The CTI synthesizer alarm TTL output has also been wired to the monitor connector, on pin D. It is accessed by connecting the white wire with the yellow tracer to the input/output board J1 connector, pin 28.

During the SBS testing in building T-80 a bug was fixed in the resident control software. When the PC software changes control between the PIFS and the SBS it sends out three consecutive '+' signs. The SBS was interpreting the last of these as a command to change the RF frequency to 9115 MHz. This was fixed by putting the line 'delay(200);' immediately before the clear_buffer() function call in the check_plus function.

A solid aluminum cylinder 1.25 inches in diameter and 36.7 cm long is used for mounting the yagi modem antenna. The yagi protrudes well above the parabolic dish and is solidly held in place in the mount on top of the SBS enclosure.

## 5.0 INSTALLATION ON THE SILO

On 30 May 1996 Grant Duff, a government employee, signed out a pickup truck for the day. He drove Manuel Dura and Gilles Caron to the silo near Luskville to install the SBS. Two long stepladders were used to get up to the bottom of the track on the silo. A small temporary pulley was attached to one of the track cross members. A rope was fed through the pulley and tied to the antenna frame. The antenna and frame were pulled up by Grant with the rope. This made it relatively easy to line it up and get the bolts in. After this was done the SBS was installed.

The rope was tied around the handles of the SBS and Grant helped pull it up while Gilles and Manuel assisted and guided it

into position. After it was bolted into place the modem antenna was installed and the connection to the parabolic antenna was made. Weather tape was used around the exposed connections. It was then plugged in. The skirt and filter was assembled and the entire package was hoisted up the side of the silo. Communication and control was tested successfully. The RF signal was received by the EARS arrays and observed at the IF monitor outputs of the Vertical Array IF receiver.

It is required by the statement of work to demonstrate the SBS to the scientific authority. In the absence of Denis Lamothe the demonstration was given to Ed Riseborough. All functions of the SBS were demonstrated with and without the EARS transmitting. When the transmitter was on the SBS was able to jam the transmitted returns quite well, particularly with noise modulation. This showed that the parabolic dish was still well aimed.

## 6.0 MAINTENANCE

The SBS has been designed to operate unattended throughout the year. It will regulate the internal temperature but does not regulate the humidity. It is recommended that it be taken off the silo and checked internally every six months, possibly in the spring and fall.

A maintenance check should consist of two things, providing the SBS is otherwise operating correctly. The metal skirt filter should be cleaned and both fan filter assemblies should be taken apart and thoroughly cleaned or the filters can simply be replaced. The second maintenance item is to open the enclosure and check for corroded or loose connections. The external connections should also be checked. Any foreign matter such as dirt or bug parts should be vacuumed out. Any accumulations of dirt or dust on the components should be cleaned off since they can interfere with cooling. Check for water infiltration and fix any physical damage that may have caused it.

## 7.0 OPERATION

The SBS is controlled in exactly the same way as the PIFS, except for the absence of the standby/operate function and the under voltage circuit. The PIFSCON program is easy to use and is user friendly. It requires a mouse and needs the modem installed

on COM port 2. This can easily be changed to COM port 1 if required. The keyboard is not used. The file 'EGAVGA.BGI' must be in the same directory as the PIFSCON.EXE program. This is the graphics driver that the program uses for the VGA monitor. A VGA monitor is required. It will run on a '386 PC or later processor.

To start the program simply type 'PIFSCON' from its directory. The program will prompt the user to connect the modem and turn it on. Press any key. Contact with PIFS 1 is attempted. If this is not successful the program continues. Use the mouse and click on any of the boxes displayed on the left side of the screen to change the parameters and send them to the PIFS or SBS. The current status of each unit is displayed on the right window. A double beep sound is heard each time the status is updated. A box at the bottom of the left window can be clicked on to request the PIFS/SBS status at any time. Click on the EXIT box to terminate the program.

Local control of the SBS is done in exactly the same way as the PIFS, using the hand-held terminal. See the previous PIFS design documents (references 1 and 2) for an explanation of the local control.

# APPENDIX A

# COMPLETE PARTS LIST

## SILO BEACON SOURCE - PARTS LIST

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| J1 | UG-30D/U | AMPHENOL | N-TYPE CONNECTOR |
| J2 | 2084-0000-02 | OMNI SPECTRA | SMA-TYPE CONNECTOR |
| J3 | KPT02E12-3P | CANNON | 3-PIN CIRCULAR BOX MOUNT CONNECTOR |
| J4 | SE-102A6-50 | SUTTLE | |
| AMP1 | AFD2-080120-55 | MITEQ | AMPLIFIER |
| AMP2 | AMF-4B-089094-80-29P | MITEQ | AMPLIFIER |
| ANT1 | 640 | NARDA | HORN ANTENNA |
| AT1 | MODEL: PYA-913 | | 900 MHz YAGI ANTENNA, 15.8 dBi |
| S1 | GHOF 4-24 | HAMMOND | 24 VDC @ 6.0 AMPS POWER SUPPLY |
| M1 | ISM900-1 | UTILICOM | ISM (900 MHz) SPREAD SPECTUM MODEM |

**POWER SUPPLY BOARD**

| | | | |
|---|---|---|---|
| R1 | MRS25F-200 | PHILIPS | 200Ω, 1%, 0.6 WATT RESISTOR |
| R2 | MRS25F-51.1 | PHILIPS | 51.1Ω, 1%, 0.6 WATT RESISTOR |
| R3 | MRS25F-200 | PHILIPS | 200Ω, 1%, 0.6 WATT RESISTOR |
| R4 | MRS25F-51.1 | PHILIPS | 51.1Ω, 1%, 0.6 WATT RESISTOR |
| R5 | PW5-15 | IRC | 15Ω, 10%, 5 WATT RESISTOR |
| R6 | MRS25F-909 | PHILIPS | 909Ω, 1%, 0.6 WATT RESISTOR |
| R7 | MRS25F-10K | PHILIPS | 10KΩ, 1%, 0.6 WATT RESISTOR |
| R8 | MRS25F-825 | PHILIPS | 825Ω, 1%, 0.6 WATT RESISTOR |
| R9 | SPH-330 | IRC | 330Ω, 5%, 3 WATT RESISTOR |
| R10 | MRS25F-681 | PHILIPS | 681Ω, 1%, 0.6 WATT RESISTOR |
| R11 | MRS25F-100 | PHILIPS | 100Ω, 1%, 0.6 WATT RESISTOR |
| R12 | MRS25F-681 | PHILIPS | 681Ω, 1%, 0.6 WATT RESISTOR |
| R13 | MRS25F-100 | PHILIPS | 100Ω, 1%, 0.6 WATT RESISTOR |
| R14 | MRS25F-49.9 | PHILIPS | 49.9Ω, 1%, 0.6 WATT RESISTOR |
| R15 | MRS25F-5K11 | PHILIPS | 5.11KΩ, 1%, 0.6 WATT RESISTOR |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| **POWER SUPPLY BOARD - Continued** | | | |
| R16 | MRS25F-5K11 | PHILIPS | 5.11KΩ, 1%, 0.6 WATT RESISTOR |
| R20 | MRS25F-5K62 | PHILIPS | 5.62KΩ, 1%, 0.6 WATT RESISTOR |
| R21 | MRS25F-1K33 | PHILIPS | 1.33KΩ, 1%, 0.6 WATT RESISTOR |
| R22 | NOT USED | | |
| R23 | MRS25F-7K5 | PHILIPS | 7.5KΩ, 1%, 0.6 WATT RESISTOR |
| R24 | MRS25F-3K01 | PHILIPS | 3.01KΩ, 1%, 0.6 WATT RESISTOR |
| R25 | MRS25F-4K75 | PHILIPS | 4.75KΩ, 1%, 0.6 WATT RESISTOR |
| R26 | MRS25F-4K75 | PHILIPS | 4.75KΩ, 1%, 0.6 WATT RESISTOR |
| R27 | 25J75 | OHMITE | 75Ω, 5%, 5 WATT RESISTOR |
| C1 | T353K226M035AS | KEMET | 22uF, 35 VDC, TANTALUM CAP. |
| C2 | T353G106M035AS | KEMET | 10uF, 35 VDC, TANTALUM CAP. |
| C3 | T353G106M035AS | KEMET | 10uF, 35 VDC, TANTALUM CAP. |
| C4 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C5 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C6 | T353K226M035AS | KEMET | 22uF, 35 VDC, TANTALUM CAP. |
| C7 | T353G106M035AS | KEMET | 10uF, 35 VDC, TANTALUM CAP. |
| C8 | T353G106M035AS | KEMET | 10uF, 35 VDC, TANTALUM CAP. |
| C9 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C10 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C11 | SME50VB101 | UNITED CHEMI-CON | 100uF, 50 VDC, ELECTROLITIC CAP. |
| C12 | SME50VB471 | UNITED CHEMI-CON | 470uF, 50 VDC, ELECTROLITIC CAP. |
| C13 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C14 | T353G106M035AS | KEMET | 10uF, 35 VDC, TANTALUM CAP. |
| C15 | T353G106M035AS | KEMET | 10uF, 35 VDC, TANTALUM CAP. |
| C16 | 035-90028 | PHILIPS | 680uF, 50 VDC, ELECTROLITIC CAP. |
| C17 | SME50VB471 | UNITED CHEMI-CON | 470uF, 50 VDC, ELECTROLITIC CAP. |
| C18 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C19 | NOT USED | | |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| **POWER SUPPLY BOARD - Continued** | | | |
| C20 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C21 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C22 | SME50VB471 | UNITED CHEMI-CON | 470uF, 50 VDC, ELECTROLITIC CAP. |
| VR1 | LM150K | NATIONAL SEMICONDUCTOR | +1.2V TO +33V, 3 AMP REGULATOR |
| VR2 | LM150K | NATIONAL SEMICONDUCTOR | +1.2V TO +33V, 3 AMP REGULATOR |
| U1 | WR24S15/2000U | COMPUTER PRODUCTS | 24V IN/15V OUT, 2 AMP DC-DC CONVERTER |
| U2 | LM120-5 | NATIONAL SEMICONDUCTOR | FIXED -5V, 1 AMP REGULATOR |
| U3 | 5VB3 | CORCOM | 5 AMP EMI/RFI FILTER |
| U4 | PH50S24-24 | LAMBDA | 24V IN/24 OUT, 2 AMP DC-DC CONVERTER |
| U5 | MAX393EP3 | MAXIM | QUAD ANALOGUE SWITCH |
| U6 | MAX393EP3 | MAXIM | QUAD ANALOGUE SWITCH |
| U7 | DAC0800LJ | NATIONAL SEMICONDUCTOR | 8 BIT DIGITAL TO ANALOGUE CONVERTER |
| U8 | LM224N | NATIONAL SEMICONDUCTOR | QUAD OP-AMP |
| U9 | MAX293EPA | MAXIM | LOW PASS IC FILTER |
| U10 | MC33164 | MOTOROLA | UNDERVOLT SENSOR |
| Q1 | 2N2222A | MOTOROLA | 40V, 0.5 AMP, NPN TRANSISTOR |
| Q2 | 2N2222A | MOTOROLA | 40V, 0.5 AMP, NPN TRANSISTOR |
| K1 | G6AK474PSTUS5VDC | OMRON | 4PDT, 2 AMP LATCHING RELAY |
| D1 | 1N4006 | MOTOROLA | 800V, 1 AMP DIODE |
| D2 | 1N4006 | MOTOROLA | 800V, 1 AMP DIODE |
| D3 | 1N4006 | MOTOROLA | 800V, 1 AMP DIODE |
| D4 | 1N4006 | MOTOROLA | 800V, 1 AMP DIODE |
| D5 | 1N4006 | MOTOROLA | 800V, 1 AMP DIODE |
| D8 | 1N4148 | MOTOROLA | 100V, 75mA DIODE |
| D9 | 1N4148 | MOTOROLA | 100V, 75mA DIODE |
| D10 | 1N4148 | MOTOROLA | 100V, 75mA DIODE |
| D11 | 1N4148 | MOTOROLA | 100V, 75mA DIODE |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| **POWER SUPPLY BOARD - Continued** | | | |
| SCR1 | MCR106-4 | MOTOROLA | 200V, 4 AMP SCR |
| SCR2 | MCR106-4 | MOTOROLA | 200V, 4 AMP SCR |
| Z1 | 1N4735A | MOTOROLA | 6.2V, 1 WATT, 5% ZENER DIODE |
| Z2 | 1N4746A | MOTOROLA | 18.0V, 1 WATT, 5% ZENER DIODE |
| POT1 | 3299W-1-102 | BOURNS | 1KΩ, 25 TURN, 0.5 WATT POT |
| POT2 | 3299W-1-502 | BOURNS | 5KΩ, 25 TURN, 0.5 WATT POT |
| POT3 | 3299W-1-103 | BOURNS | 10KΩ, 25 TURN, 0.5 WATT POT |
| POT4 | 3299W-1-102 | BOURNS | 1KΩ, 25 TURN, 0.5 WATT POT |
| J1 | 1-640445-2 | AMP | 12 POSITION HEADER |
| J2 | 640445-6 | AMP | 6 POSITION HEADER |
| J3 | 640456-8 | AMP | 8 POSITION HEADER |
| J4 | 3429-6302 | 3M | 26 POSITION HEADER |
| **DAUGHTER POWER SUPPLY BOARD** | | | |
| R1 | 615-5R-1% | RCD | 5Ω, 1%, 25 WATT RESISTOR |
| R2 | MRS25F-750 | PHILIPS | 750Ω, 1%, 0.6 WATT RESISTOR |
| R3 | MRS25F-200 | PHILIPS | 200Ω, 1%, 0.6 WATT RESISTOR |
| R4 | MRS25F-51 | PHILIPS | 51Ω, 1%, 0.6 WATT RESISTOR |
| C1 | SME50VB101M8X11LL | UNITED CHEMI-CON | 100uF, 50 VDC, ELECTROLITIC CAP. |
| C2 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C3 | T356K226K035AS | KEMET | 22uF, 35 VDC, TANTALUM CAP. |
| C4 | C322C104K1R5CA | KEMET | 0.1uF, 100 VDC, CERAMIC CAP. |
| C5 | T356K226K035AS | KEMET | 22uF, 35 VDC, TANTALUM CAP. |
| C6 | SME50VB101M8X11LL | UNITED CHEMI-CON | 100uF, 50 VDC, ELECTROLITIC CAP. |
| CR1 | 1N4006 | MOTOROLA | 800V, 1 AMP, DIODE |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| **DAUGHTER POWER SUPPLY BOARD - Continued** | | | |
| CR2 | 1N4006 | MOTOROLA | 800V, 1 AMP, DIODE |
| CR3 | 1N4738A | MOTOROLA | 8.2V, 1 WATT, 5% ZENER DIODE |
| SCR1 | MCR106-4 | MOTOROLA | 200V, 4 AMP SCR |
| U1 | LM150K | NATIONAL SEMICONDUCTOR | +1.2V TO +33V, 3 AMP REGULATOR |
| U3 | 74F163AN | MOTOROLA | PRESATTABLE 4-BIT BINARY COUNTER, SYNCHRONOUS RESET |
| U4 | 74F163AN | MOTOROLA | PRESATTABLE 4-BIT BINARY COUNTER, SYNCHRONOUS RESET |
| U5 | 74F163AN | MOTOROLA | PRESATTABLE 4-BIT BINARY COUNTER, SYNCHRONOUS RESET |
| U6 | 74F163AN | MOTOROLA | PRESATTABLE 4-BIT BINARY COUNTER, SYNCHRONOUS RESET |
| J1 | 3429-6305 | 3M | 26 PIN HEADER |

**QUADRATURE DRIVE BOARD**

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| R1 | MRS25F-649 | PHILIPS | 649Ω, 1%, 0.6 WATT RESISTOR |
| R2 | NOT USED | | |
| R3 | MRS25F-200K | PHILIPS | 200KΩ, 1%, 0.6 WATT RESISTOR |
| R4 | NOT USED | | |
| R5 | MRS25F-499 | PHILIPS | 499Ω, 1%, 0.6 WATT RESISTOR |
| R6 | MRS25F-619 | PHILIPS | 619Ω, 1%, 0.6 WATT RESISTOR |
| R7 | MRS25F-619 | PHILIPS | 619Ω, 1%, 0.6 WATT RESISTOR |
| R8 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |
| R9 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| QUADRATURE DRIVE BOARD - Continued | | | |
| R10 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |
| R11 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |
| R12 | MRS25F-13K3 | PHILIPS | 13.3KΩ, 1%, 0.6 WATT RESISTOR |
| R13 | NOT USED | | |
| R14 | NOT USED | | |
| R15 | MRS25F-402K | PHILIPS | 402KΩ, 1%, 0.6 WATT RESISTOR |
| R16 | MRS25F-249 | PHILIPS | 249Ω, 1%, 0.6 WATT RESISTOR |
| R17 | MRS25F-7K5 | PHILIPS | 7.5KΩ, 1%, 0.6 WATT RESISTOR |
| R18 | MRS25F-261K | PHILIPS | 261KΩ, 1%, 0.6 WATT RESISTOR |
| R19 | NOT USED | | |
| R20 | MRS25F-32.4 | PHILIPS | 32.4Ω, 1%, 0.6 WATT RESISTOR |
| R21 | MRS25F-1K13 | PHILIPS | 1.13KΩ, 1%, 0.6 WATT RESISTOR |
| R22 | MRS25F-37K4 | PHILIPS | 37.4KΩ, 1%, 0.6 WATT RESISTOR |
| R23 | MRS25F-619 | PHILIPS | 619Ω, 1%, 0.6 WATT RESISTOR |
| R24 | MRS25F-619 | PHILIPS | 619Ω, 1%, 0.6 WATT RESISTOR |
| R25 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |
| R26 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |
| R27 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |
| R28 | MRS25F-562 | PHILIPS | 562Ω, 1%, 0.6 WATT RESISTOR |
| R29 | MRS25F-191 | PHILIPS | 191Ω, 1%, 0.6 WATT RESISTOR |
| R30 | MRS25F-4K64 | PHILIPS | 4.64KΩ, 1%, 0.6 WATT RESISTOR |
| R31 | MRS25F-107K | PHILIPS | 107KΩ, 1%, 0.6 WATT RESISTOR |
| R32 | MRS25F-95.3 | PHILIPS | 95.3Ω, 1%, 0.6 WATT RESISTOR |
| R33 | MRS25F-6.04 | PHILIPS | 6.04Ω, 1%, 0.6 WATT RESISTOR |
| R34 | MRS25F-3.57 | PHILIPS | 3.57Ω, 1%, 0.6 WATT RESISTOR |
| R35 | MRS25F-210 | PHILIPS | 210Ω, 1%, 0.6 WATT RESISTOR |
| R36 | MRS25F-2K43 | PHILIPS | 2.43KΩ, 1%, 0.6 WATT RESISTOR |
| R37 | MRS25F-10K | PHILIPS | 10KΩ, 1%, 0.6 WATT RESISTOR |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| QUADRATURE DRIVE BOARD - Continued | | | |
| R38 | MRS25F-66K5 | PHILIPS | 66.5KΩ, 1%, 0.6 WATT RESISTOR |
| R39 | MRS25F-11K | PHILIPS | 11KΩ, 1%, 0.6 WATT RESISTOR |
| R40 | MRS25F-10K | PHILIPS | 10KΩ, 1%, 0.6 WATT RESISTOR |
| R41 | MRS25F-127 | PHILIPS | 127Ω, 1%, 0.6 WATT RESISTOR |
| R42 | MRS25F-1M | PHILIPS | 1MΩ, 1%, 0.6 WATT RESISTOR |
| R43 | MRS25F-2 | PHILIPS | 2Ω, 1%, 0.6 WATT RESISTOR |
| R44 | MRS25F-49.9 | PHILIPS | 49.9Ω, 1%, 0.6 WATT RESISTOR |
| R45 | MRS25F-49.9 | PHILIPS | 49.9Ω, 1%, 0.6 WATT RESISTOR |
| R46 | MRS25F-649 | PHILIPS | 649Ω, 1%, 0.6 WATT RESISTOR |
| C1 | 424-32702 | PHILIPS | 2.7nF CAP., 2%, 63 VDC |
| C2 | 424-32702 | PHILIPS | 2.7nF CAP., 2%, 63 VDC |
| C3 | 424-31003 | PHILIPS | 10nF CAP., 2%, 63 VDC |
| C4 | 424-31303 | PHILIPS | 13nF CAP., 2%, 63 VDC |
| C5 | 424-33903 | PHILIPS | 39nF CAP., 2%, 63 VDC |
| C6 | 424-33903 | PHILIPS | 39nF CAP., 2%, 63 VDC |
| C7 | 424-32002 | PHILIPS | 2nF CAP., 2%, 63 VDC |
| C8 | 424-32002 | PHILIPS | 2nF CAP., 2%, 63 VDC |
| C9 | 424-31003 | PHILIPS | 10nF CAP., 2%, 63 VDC |
| C10 | 424-31003 | PHILIPS | 11nF CAP., 2%, 63 VDC |
| C11 | 424-33903 | PHILIPS | 39nF CAP., 2%, 63 VDC |
| C12 | 424-33903 | PHILIPS | 39nF CAP., 2%, 63 VDC |
| C13 | CK05BX104M | KEMET | 0.1uF CAP., 20%, 50 VDC |
| C14 | CN20A152J | CENTRALAB | 1.5nF CAP., 5%, 100 VDC |
| C15 | 424-33603 | PHILIPS | 36nF CAP., 2%, 63 VDC |
| C16 | 424-32403 | PHILIPS | 24nF CAP., 2% 63 VDC |
| C17 | 681 09478 | PHILIPS | 4.7pF CAP., 2%, 100 VDC |
| U1 | AD811AN | ANALOG DEVICES | VIDEO CURRENT FEEDBACK OP-AMP |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| QUADRATURE DRIVE BOARD - Continued | | | |
| U2 | AD811AN | ANALOG DEVICES | VIDEO CURRENT FEEDBACK OP-AMP |
| U3 | AD811AN | ANALOG DEVICES | VIDEO CURRENT FEEDBACK OP-AMP |
| U4 | AD811AN | ANALOG DEVICES | VIDEO CURRENT FEEDBACK OP-AMP |
| U5 | AD811AN | ANALOG DEVICES | VIDEO CURRENT FEEDBACK OP-AMP |
| U6 | AD811AN | ANALOG DEVICES | VIDEO CURRENT FEEDBACK OP-AMP |
| U7 | AD811AN | ANALOG DEVICES | VIDEO CURRENT FEEDBACK OP-AMP |
| TEMPERATURE CONTROL ASSEMBLY | | | |
| FAN1 | MX2A3 | MUFFIN/COMAIR ROTRON | 115 CFM, 3100 RPM, 0.18 AMP, SLEEVE |
| FAN2 | MX2A3 | MUFFIN/COMAIR ROTRON | 115 CFM, 3100 RPM, 0.18 AMP, SLEEVE |
| R1 | 615-30R-1% | RCD | 30Ω, 1%, 25 WATT RESISTOR |
| R2 | 615-30R-1% | RCD | 30Ω, 1%, 25 WATT RESISTOR |
| R3 | 615-30R-1% | RCD | 30Ω, 1%, 25 WATT RESISTOR |
| R4 | 615-30R-1% | RCD | 30Ω, 1%, 25 WATT RESISTOR |
| R5 | 615-30R-1% | RCD | 30Ω, 1%, 25 WATT RESISTOR |
| R6 | RH-5-20Ω | DALE | 20Ω, 1%, 5 WATT RESISTOR |
| R7 | RH-5-20Ω | DALE | 20Ω, 1%, 5 WATT RESISTOR |
| R8 | RH-5-20Ω | DALE | 20Ω, 1%, 5 WATT RESISTOR |
| R9 | RH-5-20Ω | DALE | 20Ω, 1%, 5 WATT RESISTOR |
| R10 | 615-100R-1% | RCD | 100Ω, 1%, 25 WATT RESISTOR |
| R11 | 615-100R-1% | RCD | 100Ω, 1%, 25 WATT RESISTOR |
| S1 | STC-120 | STANCOR | THERMAL SWITCH, CLOSES ON TEMPERATURE RISE, 120°F (50°C) |
| S2 | TO-35-QC | RYTON | THERMAL SWITCH, CLOSE ON TEMPERATURE DECREASE, 35°F (1.6°C) |

| LABEL | PART NUMBER | MANUFACTURER | DESCRIPTION |
|---|---|---|---|
| S3 | TO-35-QC | RYTON | THERMAL SWITCH, CLOSE ON TEMPERATURE DECREASE, 35°F (1.6°C) |

# APPENDIX B

# RESIDENT CONTROL SOFTWARE SOURCE LISTING

```
/*This program will control the components in the remote controlled Silo
  Beacon Source (SBS).  It is downloaded in Octagon's single board
  computer's flash non-volatile memory.
  Written by Gilles Caron, Applied Silicon Inc. Canada
  Modified from the PIFS2 control program for control of the SBS.
  Modified on 13 May 1996 by John Curnow  */

#include "async2.h"
#include <conio.h>
#include <bios.h>
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <stdlib.h>

#define BASE_ADDR   0x100
#define _8MHz       8000000.0

void status_display(void);
void test_alarms(void);
void local_mode(void);
void remote_mode(void);
void local_control(void);
void get_char_remote(void);
void get_char_remote2(void);
void check_for_status(void);
void check_plus(void);
void get_char(void);
void get_char2(void);
void interpret_remote_command(void);
void set_RF_frequency(void);
void set_attenuation(void);
void set_RF_enable(void);
void set_modulation(void);
void send_status(void);
void set_doppler_frequency(void);
void clear_buffer(void);
void clear_buffer2(void);

int loc_rem_bit, loc_rem_test,
    flagtest, temp_int, flag = 0, flag2 = 0,
    attenuation_int = 0, RF_freq_int = 8900,
    transmit_enable = 1, err, modulation=1,
    byte_1 = 0, byte_2 = 0, byte_3 = 0, byte_4 = 0;

unsigned int least_5 = 0, most_8 = 0, mod_temp = 0, op_st_dc_dn = 8,
          command = 0, RF_frequency = 0, atten_RF_en = 1, CTI_alarm=0;

char remote_command, local_command, RF_freq_ascii[4],
     temp_char1[2], temp_char2[3], temp_char3[4], temp_char4[5],
     attenuation_char[3], doppler_frequency_char[6], message1 [ ] = "ATD3",
     flush, recheck, receive_ch, character[2], enter = 0x0D;

unsigned long int command_temp, counter_clock, doppler_frequency_int = 0,
             doppler_temp = 0, counter_0_value_int, counter_1_value_int,
             temp2_int, filter_clock, counter_0_value, counter_1_value;
```

```
main()
{
  // first initialize the two COM ports
  // COM 1 is for the modem, COM 2 is for the terminal
  sethandshakemode(0,0);                  // set port 1 no handshaking
  sethandshakemode(1,0);                  // set port 2 no handshaking
  opencom(0,9600,0,1,8,0);                // set port 1 9600,8,N,1
  opencom(1,9600,0,1,8,0);                // set port 2 9600,8,N,1
  delay(2000);
  writecom(0,message1,0);
  sendcom(0, enter, 0);
  delay(2000);
  clear_buffer();
  clear_buffer2();
  // set port A as output, B as output
  // port C lower 4 bits are output and upper 4 bits are input
  outp(BASE_ADDR + 0x03, 0x88);
  outp(BASE_ADDR + 0x00, RF_frequency); // send the default parameters
  outp(BASE_ADDR + 0x01, atten_RF_en);  // to the IO lines
  outp(BASE_ADDR + 0x02, op_st_dc_dn);
  writecom(1,"\x1B[?8l",&err);  // disable key repeat
  writecom(1,"\x1B[0t",&err);   // enable key click
  writecom(1,"\x1B[3t",&err);   // disable Knp function
  writecom(1,"\x1B[4t",&err);   // enable cursor
  writecom(1,"\x1B[6t",&err);   // enable cursor blink
  writecom(1,"\x1B[5zf1;52\x1B[0z",&err);  // F1 = 'R' for RF ON/OFF
  delay(100);           // allow time for the 8045 to program the function key
  writecom(1,"\x1B[5zf2;46\x1B[0z",&err);  // F2 = 'F' for RF frequency
  delay(100);
  writecom(1,"\x1B[5zf3;41\x1B[0z",&err);  // F3 = 'A' for attenuation
  delay(100);
  writecom(1,"\x1B[5zf4;54\x1B[0z",&err);  // F4 = 'T' for modulation type
  delay(100);
  writecom(1,"\x1B[5zf5;44\x1B[0z",&err);  // F5 = 'D' for doppler frequency
  delay(75);
  loc_rem_bit = inp(BASE_ADDR + 0x02);
  if(loc_rem_bit & 0x40) writecom(0,"L",0); // send 'L' for local control
  else writecom(0,"R",0);          // send 'R' as ready message to controller
  do
  {
    loc_rem_bit = inp(BASE_ADDR + 0x02);
  } while((loc_rem_bit & 0x20) == 0); // wait until CTI alarm bit is no alarm
  status_display();
  loc_rem_bit = inp(BASE_ADDR + 0x02);  // input the bit for local or remote status
  if(loc_rem_bit & 0x40) flag = 1;      // 1 = local
  else flag = 0;                        // 0 = remote
  test_alarms();
  return(0);
}
```

```
void test_alarms(void)
{
  loc_rem_bit = inp(BASE_ADDR + 0x02);
  delay(100);
  if((loc_rem_bit & 0x20) == 0)  // CTI alarm has happened (low bit)
  {
    writecom(0,"A",0);     // send CTI alarm message to controller
    command = 167;
    CTI_alarm = 64;
  }
  else CTI_alarm=0;
  flagtest = (loc_rem_bit & 0x40);  // check local/remote switch input
  flagtest = flagtest >> 6;
  if(flagtest == flag)
  {
    if(loc_rem_bit & 0x40)
    {
      checkcom(0,&receive_ch,&err);
      if(err == 0) check_for_status();
      local_mode();
    }
    remote_mode();
  }

  if(flagtest != flag)
  {
    delay(500);
    if(loc_rem_bit & 0x40)
    {
      checkcom(0,&receive_ch,&err);
      if(err == 0) check_for_status();
      clear_buffer2();
      local_mode();
    }
    clear_buffer();
    remote_mode();
  }
  return;
}

void remote_mode(void)
{
  if(flag == 1)
  {
    writecom(0,"R",0); // send 'R' as remote message to controller
    flag = 0;
  }
  get_char_remote();
  return;
}
void get_char_remote(void)
{
  checkcom(0,&remote_command,&err);
  if(err != 0) test_alarms();
  interpret_remote_command();
  return;
}
```

```
void interpret_remote_command(void)       // this section converts the
{                                         // ASCII character received
  command = 0;                            // over the serial port to an
                                  // integer one bit at a time
  if(remote_command & 0x80) command = (command | 0x80);
  if(remote_command & 0x40) command = (command | 0x40);
  if(remote_command & 0x20) command = (command | 0x20);
  if(remote_command & 0x10) command = (command | 0x10);
  if(remote_command & 0x08) command = (command | 0x08);
  if(remote_command & 0x04) command = (command | 0x04);
  if(remote_command & 0x02) command = (command | 0x02);
  if(remote_command & 0x01) command = (command | 0x01);
  if(command <= 0x64 && flag!=1)            // remote command
  {
    if(command == 43) check_plus();
    set_RF_frequency();
  }
  if(0x65 <= command && command <= 0xA4 && flag!=1) set_attenuation();
  if(0xA5 <= command && command <= 0xA6 && flag!=1) set_RF_enable();
  if(0xA9 <= command && command <= 0xAB && flag!=1) set_modulation();
  if(command == 0xAC) send_status(); // status can be sent in local or remote
  if(command >= 0xE0 && flag!=1)
  {
    command = (command & 0x1F);
    least_5 = 0;
    least_5 = command;
    do
    {
      checkcom(0,&receive_ch,&err);
    } while(err != 0);
    temp2_int = 0;       // reset the temporary integer
    if(receive_ch & 0x80) temp2_int = (temp2_int | 0x80);
    if(receive_ch & 0x40) temp2_int = (temp2_int | 0x40);
    if(receive_ch & 0x20) temp2_int = (temp2_int | 0x20);
    if(receive_ch & 0x10) temp2_int = (temp2_int | 0x10);
    if(receive_ch & 0x08) temp2_int = (temp2_int | 0x08);
    if(receive_ch & 0x04) temp2_int = (temp2_int | 0x04);
    if(receive_ch & 0x02) temp2_int = (temp2_int | 0x02);
    if(receive_ch & 0x01) temp2_int = (temp2_int | 0x01);
    most_8 = temp2_int;
    temp2_int = (temp2_int << 5);
    doppler_frequency_int = 0;      // reset the doppler frequency
    doppler_frequency_int = (temp2_int | command);
    set_doppler_frequency();
  }
  test_alarms();
  return;
}

void status_display(void)
{
  delay(200);
  writecom(1,"\x1B[1s",&err);  // erase entire display and send cursor home
  writecom(1,"TX",&err);
  if (transmit_enable == 1) writecom(1," ON      ",&err);
  else writecom(1," OFF     ",&err);
  itoa(RF_freq_int, RF_freq_ascii, 10);
```

```
    strncpy(temp_char1, RF_freq_ascii, 1);
    writecom(1,temp_char1,&err);              // write the MHz thousands value
    writecom(1,".",&err);
    strncpy(temp_char2, RF_freq_ascii, 2);  // copy the first two digits
    strrev(temp_char2);                       // reverse the string
    strncpy(temp_char1, temp_char2, 1);
    writecom(1,temp_char1,&err);              // write the MHz hundreds value
    strncpy(temp_char3, RF_freq_ascii, 3);  // copy the first two digits
    strrev(temp_char3);                       // reverse the string
    strncpy(temp_char1, temp_char3, 1);
    writecom(1,temp_char1,&err);              // write the MHz tens value
    strncpy(temp_char4, RF_freq_ascii, 4);  // copy the first two digits
    strrev(temp_char4);                       // reverse the string
    strncpy(temp_char1, temp_char4, 1);
    writecom(1,temp_char1,&err);              // write the MHz tens value
    writecom(1," GHz",&err);
    writecom(1,"ATT=",&err);
    itoa(attenuation_int, attenuation_char, 10);
    if (attenuation_int > 9) writecom(1,attenuation_char,&err);
    else
    {
      writecom(1," ",&err);
      writecom(1,attenuation_char,&err);
    }
    writecom(1," dB ",&err);
    switch (modulation)
    {
      case 1: {
        writecom(1,"  NO MODUL",&err);
        break;
      }
      case 2: {
        writecom(1," NOISE MOD",&err);
        break;
      }
      case 3: {
        writecom(1,"DOPPLR MOD",&err);
        break;
      }
    }
    writecom(1,"DOPP= ",&err);
    itoa(doppler_frequency_int,doppler_frequency_char,10);
    if (doppler_frequency_int < 10000) writecom(1," ",&err);
    if (doppler_frequency_int < 1000) writecom(1," ",&err);
    if (doppler_frequency_int < 100) writecom(1," ",&err);
    if (doppler_frequency_int < 10) writecom(1," ",&err);
    writecom(1,doppler_frequency_char,&err);
    writecom(1," Hz",&err);
    writecom(1,"\x1B[4;1f",&err);  // position cursor to start of last row
    writecom(1,"M=FUNCTION KEY MENU",&err);
    delay(100);
    return;
}

void menu_display(void)
{
    writecom(1,"\x1B[1s",&err);  // erase entire display and send cursor home
```

```
  writecom(1,"F1=RF OUTPUT ON/OFF ",&err);
  writecom(1,"F2=RF FREQUENCY     ",&err);
  writecom(1,"F3=ATTEN. F4=MODULA.",&err);
  writecom(1,"F5=DOPPLER FREQ.",&err);
  delay(3000);                  // wait for three seconds
  return;
}

void clear_buffer(void)
{
  do
  {
    checkcom(0,&flush,&err);
  } while(err != 6);
  return;
}

void clear_buffer2(void)
{
  do
  {
    checkcom(1,&flush,&err);
  } while(err != 6);
  return;
}

void get_digit(void)
{
  get_char2();
  character[0] = receive_ch;
  temp_int = atoi(character);
  return;
}

void get_char(void) // get a character from the com port until the buffer is empty
{
  checkcom(1,&receive_ch,&err);
  if(err != 0) test_alarms();
  return;
}

void get_char2(void)
{
  do
  {
    checkcom(1,&receive_ch,&err);
  } while(err != 0);
  return;
}

void get_char_remote2(void)
{
  do
  {
    checkcom(0,&remote_command,&err);
  } while(err != 0);
  if(remote_command == '¼') send_status();
```

```
  do
  {
    checkcom(0,&flush,&err);
  } while(err != 6);
  get_char_remote2();
  return;
}

void check_for_status(void)
{
  if(receive_ch == '¼') // request for status
  {
    send_status();
    return;
  }
  clear_buffer();
  return;
}

void local_mode(void)
{
  if(flag == 0)
  {
    writecom(0,"L",0);
    flag = 1;
  }
  local_control();
  return;
}

void local_control(void)
{
  get_char();              // wait for character from 8045 terminal
  switch (receive_ch)
  {
    case 'M': {            // M means to display the menu for a short time
      menu_display();
      status_display();
      break;
    }
    case 'R': {            // R toggles the RF output on/off
      writecom(1,"\x1B[1;4H",&err);  // position cursor
      if (transmit_enable == 0)
      {
      writecom(1,"ON ",&err);
      transmit_enable = 1;
      command = 0xA5;
      }
      else
      {
      writecom(1,"OFF",&err);
      transmit_enable = 0;
      command = 0xA6;
      }
      writecom(1,"\x1B[4;20H",&err);  // position cursor to lower right
      set_RF_enable();
      break;
```

```
    }
    case 'F': {                          // F will set the RF frequency
      writecom(1,"\x1B[1;12H",&err);  // position cursor on GHz ones
      do
      {
      get_digit();                        // wait for GHz ones input
      if (temp_int < 8) writecom(1,"\x1B[0q",&err);  // sound short bell
      } while ( temp_int < 8 );
      RF_freq_int = 1000*temp_int;
      writecom(1,character,&err);
      writecom(1,".",&err);
      do
      {
      get_digit();                        // wait for MHz hundreds input
      if (((RF_freq_int == 9000) && (temp_int > 4)) || ((RF_freq_int == 8000) && (temp_in
!= 9))) writecom(1,"\x1B[0q",&err);  // sound short bell
        } while (((RF_freq_int == 9000) && (temp_int > 4)) || ((RF_freq_int == 8000) &
(temp_int != 9)));
      RF_freq_int = RF_freq_int + 100*temp_int;
      writecom(1,character,&err);
      do
      {
      get_digit();                        // wait for MHz tens input
      if ((RF_freq_int == 9400) && (temp_int > 0)) writecom(1,"\x1B[0q",&err);  // soun
short bell
      } while ((RF_freq_int == 9400) && (temp_int > 0));
      RF_freq_int = RF_freq_int + 10*temp_int;
      writecom(1,character,&err);
      do
      {
      get_digit();                        // wait for MHz ones input
      if (((temp_int != 0) && (temp_int != 5)) || ((RF_freq_int == 9400) && (temp_int != 0))
writecom(1,"\x1B[0q",&err);  // sound short beli
      } while (((temp_int != 0) && (temp_int != 5)) || ((RF_freq_int == 9400) && (temp_int
!= 0)));
      RF_freq_int = RF_freq_int + temp_int;
      writecom(1,character,&err);
      command = ((RF_freq_int - 8900)/5);

      writecom(1,"\x1B[4;20H",&err);  // position cursor to lower right
      set_RF_frequency();
      break;
    }
    case 'A': {            // A sets the attenuation
      writecom(1,"\x1B[2;5H",&err);  // position cursor on attenuation tens
      do
      {
      get_digit();                        // wait for attenuation tens input
      if ( temp_int > 6 ) writecom(1,"\x1B[0q",&err);  // sound short bell
      } while ( temp_int < 0 || temp_int > 6 );
      attenuation_int = 10 * temp_int;
      if (temp_int == 0) writecom(1," ",&err);          // put in blank space instead of '0'
      else writecom(1,character,&err);
      do
      {
      get_digit();                        // wait for attenuation ones input
      if ( (temp_int > 3) && (attenuation_int == 60) || temp_int < 0 || temp_int > 9)
```

```
writecom(1,"\x1B[0q",&err);  // sound short bell
      } while ( (temp_int > 3) && (attenuation_int == 60) );
      writecom(1,character,&err);
      attenuation_int = attenuation_int + temp_int;
      command = (attenuation_int + 101);

      writecom(1,"\x1B[4;20H",&err);  // position cursor to lower right
      set_attenuation();
      break;
    }
    case 'T': {            // T sets the modulation type
      writecom(1,"\x1B[2;11H",&err);  // position cursor on modulation type
      modulation++;                   // increment modulation type
      if (modulation == 4) modulation = 1;
      switch (modulation)             // change modulation type display
      {
      case 1: {
        writecom(1,"  NO MODUL",&err);
        break;
      }
      case 2: {
        writecom(1,"NOISE MOD ",&err);
        break;
      }
      case 3: {
        writecom(1,"DOPPLR MOD",&err);
        break;
      }

      }
      command = (modulation + 168);
      writecom(1,"\x1B[4;20H",&err);  // position cursor to lower right
      set_modulation();
      break;
    }

    case 'D': {            // D is for setting the doppler frequency
      writecom(1,"\x1B[3;8H",&err);  // position cursor on doppler frequency
      do
      {
      get_digit();                   // wait for doppler 1,000 input
      if (temp_int > 7) writecom(1,"\x1B[0q",&err);  // sound short bell
      } while (temp_int > 7);
      if ( temp_int == 0) writecom(1," ",&err);  // put in blank space
      else writecom(1,character,&err);
      doppler_frequency_int = 1000 * temp_int;
      do
      {
      get_digit();                   // wait for doppler hundreds input
      if (doppler_frequency_int == 7000 && temp_int == 9) writecom(1,"\x1B[0q",&err);  /,
sound short bell
      } while (doppler_frequency_int == 7000 && temp_int == 9);
       if ( temp_int == 0 && doppler_frequency_int == 0) writecom(1," ",&err);  // put ir
blank space
      else writecom(1,character,&err);
      doppler_frequency_int = doppler_frequency_int + 100 * temp_int;
      do
```

```
        {
        get_digit();                    // wait for doppler tens input
        if (doppler_frequency_int == 7800 && temp_int > 2) writecom(1,"\x1B[0q",&err);  /
 sound short bell
        } while (doppler_frequency_int == 7800 && temp_int > 2);
         if ( temp_int == 0 && doppler_frequency_int == 0) writecom(1," ",&err);  // put i
 blank space
        else writecom(1,character,&err);
        doppler_frequency_int = doppler_frequency_int + 10 * temp_int;
        do
        {
        get_digit();                    // wait for doppler ones input
        if (doppler_frequency_int == 7810 && temp_int > 3) writecom(1,"\x1B[0q",&err);  /
 sound short bell
        } while (doppler_frequency_int == 7810 && temp_int > 3);

        writecom(1,character,&err);
        writecom(1," Hz      ",&err);
        doppler_frequency_int = doppler_frequency_int + temp_int;
        doppler_temp = doppler_frequency_int;
        least_5 = doppler_temp & 0x1F;
        most_8 = doppler_temp >> 5;
        most_8 = most_8 & 0xFF;
        writecom(1,"\x1B[4;20H",&err);  // position cursor to lower right
        set_doppler_frequency();
        break;
      }
    }
    get_char_remote();  // check if request for status came through
    test_alarms();
    return;
}

void check_plus(void)
{
    do
    {
      checkcom(0,&recheck,&err);
    } while(err != 0);
    if(recheck == '+')
    {
      flag2 = 1;
      set_RF_frequency();
      flag2 = 0;
      command = 172;
      send_status();
    }
    flag2 = 0;
    delay(200);
    clear_buffer();
    test_alarms();
    return;
}

void set_RF_frequency(void)
{
    RF_frequency = 0;
```

```
    command = (command * 5);
    RF_freq_int = (command + 8900);
    if(command >= 400)
    {
      RF_frequency = (RF_frequency | 0x80);
      command = (command - 400);
    }
    if(command >= 200)
    {
      RF_frequency = (RF_frequency | 0x40);
      command = (command - 200);
    }
    if(command >= 100)
    {
      RF_frequency = (RF_frequency | 0x20);
      command = (command - 100);
    }
    if(command >= 80)
    {
      RF_frequency = (RF_frequency | 0x10);
      command = (command - 80);
    }
    if(command >= 40)
    {
      RF_frequency = (RF_frequency | 0x08);
      command = (command - 40);
    }
    if(command >= 20)
    {
      RF_frequency = (RF_frequency | 0x04);
      command = (command - 20);
    }
    if(command >= 10)
    {
      RF_frequency = (RF_frequency | 0x02);
      command = (command - 10);
    }
    if(command >= 5)
    {
      RF_frequency = (RF_frequency | 0x01);
      command = (command - 5);
    }
    outp(BASE_ADDR + 0x00, RF_frequency);       // set the output lines to
    status_display();
    delay(100);
    if(flag2 == 1) return;
    test_alarms();                              // the desired RF_frequency
    return;
}

void set_attenuation(void)
{
  unsigned int attenuation = 0;
  command = (command - 0x65);
  attenuation_int = command;
  if(command >= 32)
  {
```

```
      attenuation = (attenuation | 0x40);
      command = (command - 32);
    }
    if(command >= 16)
    {
      attenuation = (attenuation | 0x20);
      command = (command - 16);
    }
    if(command >= 8)
    {
      attenuation = (attenuation | 0x10);
      command = (command - 8);
    }
    if(command >= 4)
    {
      attenuation = (attenuation | 0x08);
      command = (command - 4);
    }
    if(command >= 2)
    {
      attenuation = (attenuation | 0x04);
      command = (command - 2);
    }
    if(command >= 1)
    {
      attenuation = (attenuation | 0x02);
      command = (command - 1);
    }
    atten_RF_en = (atten_RF_en & 0x01);
    atten_RF_en = (atten_RF_en | attenuation);
    outp(BASE_ADDR + 0x01, atten_RF_en);       // set the output lines to
    status_display();
    delay(100);
    test_alarms();                             // the desired attenuation
    return;
}

void set_RF_enable(void)
{
  if(command == 0xA5)   // enable the RF output
  {
    atten_RF_en = (atten_RF_en & 0xFE);
    transmit_enable = 1;
  }
  if(command == 0xA6)   // disable the RF output
  {
    atten_RF_en = (atten_RF_en | 0x01);
    transmit_enable = 0;
  }
  outp(BASE_ADDR + 0x01, atten_RF_en);       // set the RF_enable switch
  status_display();
  delay(100);
  test_alarms();                             // to on or off
  return;
}

void set_modulation(void) // lower 2 LSBs for modulation selection
```

```
{
  if(command == 0xA9)
  {
    op_st_dc_dn = (op_st_dc_dn & 0xFC); // clear lower two bits
    mod_temp = op_st_dc_dn;             // for CW operation or
    mod_temp = mod_temp & 0x03;         // no modulation
    mod_temp = mod_temp << 6;
    modulation = 1;
  }
  if(command == 0xAA)
  {
    op_st_dc_dn = (op_st_dc_dn & 0xFC);
    op_st_dc_dn = (op_st_dc_dn | 0x02); // set lower two bits to 10
    mod_temp = op_st_dc_dn;             // for noise modulation
    mod_temp = mod_temp & 0x03;
    mod_temp = mod_temp << 6;
    modulation = 2;
  }
  if(command == 0xAB)
  {
    op_st_dc_dn = (op_st_dc_dn & 0xFC);
    op_st_dc_dn = (op_st_dc_dn | 0x03); // set lower two bits to 11
    mod_temp = op_st_dc_dn;             // for doppler modulation
    mod_temp = mod_temp & 0x03;
    mod_temp = mod_temp << 6;
    modulation = 3;
  }
  outp(BASE_ADDR + 0x02, op_st_dc_dn);  // set output lines to the
  status_display();
  delay(100);
  test_alarms();                        // desired modulation
  return;
}

void set_doppler_frequency(void)

{  // function to set the square wave clocks to the 74ALS163 counters and
   // the MAX293 low pass filter.  The counter clock is 512*doppler_frequency_int
   // and the filter clock is 130*doppler_frequency_int. Counter 0 is used for
   // the filter, counter 1 is used for the counters for the EPROM address.
   // address of counter control register is 271 (0x100+12+3)
   // address of counter 0 data is 268 (0x100+12+0)
   // address of counter 1 data is 269 (0x100+12+1)
   // the 5300 board must have been modified to make counter 1 independent
   // and bring its output out to J1-21 in place of CTC302.
   // doppler_frequency_int is a global unsigned int variable representing
   // the doppler frequency from 10 Hz to 7812 Hz.

  counter_clock = 0;
  filter_clock = 0;
  counter_0_value = 0;
  counter_0_value_int = 0;
  counter_1_value = 0;
  counter_1_value_int = 0;

  counter_clock = 512 * doppler_frequency_int; // calculate frequency for DAC
  filter_clock =  120 * doppler_frequency_int; // low pass filter frequency is
```

```
                                        //1.3 times the doppler frequency
    counter_0_value = _8MHz / filter_clock; // get divisor for counter 0 and
    counter_0_value_int = counter_0_value;  // round it off to the nearest integer
    if ((counter_0_value-counter_0_value_int) > 0.499) counter_0_value_int++;
    counter_1_value = _8MHz / counter_clock; // get divisor for counter 1 and
    counter_1_value_int = counter_1_value;   // round it off to the nearest integer
    if ((counter_1_value-counter_1_value_int) > 0.499) counter_1_value_int++;
    outportb(271,54);     // write control word for counter 0= 00110110 (mode 3)
    outportb(268,(counter_0_value_int & 0xFF));      // write counter 0 LSB
    counter_0_value_int = counter_0_value_int >> 8; // shift right 8 bits
    outportb(268,counter_0_value_int);               // write counter 0 MSB
    outportb(271,118);    // write control word for counter 1= 01110110 (mode 3)
    outportb(269,(counter_1_value_int & 0xFF));      // write counter 1 LSB
    counter_1_value_int = counter_1_value_int >> 8; // shift right 8 bits
    outportb(269,counter_1_value_int);              // write counter 1 MSB
    status_display();
    delay(100);
    test_alarms();
    return;
}

void send_status(void)
{
    byte_1 = RF_frequency;
    sendcom(0, byte_1, 0);
    byte_2 = 0;
    byte_2 = atten_RF_en;
    byte_2 = byte_2 >> 1;
    byte_2 = byte_2 | CTI_alarm;    // bit 6 is high if synthesizer alarm is low
    if(atten_RF_en & 0x01) byte_2 = byte_2 & 0x7F;
    else byte_2 = byte_2 | 0x80;  // bit 7 is high if RF output is on
    sendcom(0, byte_2, 0);
    byte_3 = mod_temp;
    byte_3 = byte_3 & 0xC0;
    if(flag == 1) byte_3 = byte_3 | 0x20;
    if(flag == 0) byte_3 = byte_3 & 0xDF;
    byte_3 = byte_3 | least_5;
    sendcom(0, byte_3, 0);
    byte_4 = most_8;
    sendcom(0, byte_4, 0);
    test_alarms();
    return;
}
```

## REFERENCES


1. Curnow, John, *HARDWARE DESIGN SPECIFICATION DOCUMENT OF THE SILO BEACON SOURCE*, Applied Silicon Inc. Canada, May 1996, rev 1.0, Document Number 974-55126-11

2. Curnow, John, ***PORTABLE INTERFERENCE FREQUENCY SOURCES*** - *REMOTE CONTROL UPGRADE DESIGN SPECIFICATION DOCUMENT*, Applied Silicon Inc. Canada, July 1995, rev 1.0, Document Number 961-55126-02

3. Curnow, John, ***PORTABLE INTERFERENCE FREQUENCY SOURCES*** - *REMOTE CONTROL UPGRADE FINAL REPORT*, Applied Silicon Inc. Canada, 21 August 1995, rev 1.0, Document Number 974-55126-02

## ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| CRC | Communications Research Center |
| CW | Continuous Wave (no modulation) |
| EARS | Experimental Array Radar System |
| EMI | Electro-Magnetic Interference |
| DREO | Defence Research Establishment Ottawa |
| IF | Intermediate Frequency |
| PIFS | Portable Interference Frequency Source |
| RF | Radio Frequency |
| SBC | Single Board Computer |
| SBS | Silo Beacon Source |

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

## DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

APPLIED SILICON INC., CANADA
220-2427 HOLLY LANE
OTTAWA, ONTARIO KIV 7P2

2. SECURITY CLASSIFICATION (overall security classification of the document. including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

FINAL REPORT OF THE SILO BEACON SOURCE

4. AUTHORS (Last name, first name, middle initial)

CURNOW, JOHN W.

5. DATE OF PUBLICATION (month and year of publication of document)

30 MAY 1996

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

44

6b. NO. OF REFS (total cited in document)

3

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

FINAL TECHNICAL REPORT

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DND - DEFENCE RESEARCH ESTABLISHMENT OTTAWA
RADAR AND SPACE DIVISION (SURFACE RADAR)
SHIRLEY'S BAY, OTTAWA, ONTARIO

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

EARS RADAR RESEARCH PROJECT

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

W7714-5-9868

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

974-55126-12

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

DRES CONTRACTOR REPORT 98-629

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availabilty (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNLIMITED

UNCLASSIFIED

13. ABSTRACT ( a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both offical languages unless the text is bilingual).

This document presents the test results after the construction of the remote controlled frequency agile silo beacon RF source (SBS) and the changes made since the design specification document was written. The 'silo' is located near Luskville, Quebec, and is 13.5 kilometers away from the Experimental Array Radar System (EARS) at the Area Five radar test site. One of two Portable Interference Frequency Sources (PIFS) was dismantled to supply the parts for the silo source. PIFS 1 is still intact and operational. The silo beacon source has the same functionality as the PIFS, including; RF output from 8.9 to 9.4 gigahertz, CW, noise or doppler modulation, doppler frequency from one to 7813 hertz. The following functions are remotely controlled via a spread spectrum RF modem link; RF on/off, RF frequency, modulation selection, doppler frequency, RF transmit signal amplitude. A personal computer at the Area Five site runs a control program to communicate with and control the silo source. The source can also be run in local testing mode using a portable terminal. Several key design modifications were implemented to convert the PIFS hardware for use as the SBS. A 24 VDC power supply was added to replace the two gell-cell batteries that powered the PIFS. The 'standby' and low voltage shutdown modes were eliminated. Two AC fans and a heater, all under thermostatic control have been added. The resident and control software was modified slightly. A yagi antenna for the modem link has been mounted on the SBS enclosure. The new SBS enclosure was made very weather resistant, with ventilation filters and no exposed switches or connectors. It is much larger than the PIFS enclosure but is the same type of box.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

| | |
|---|---|
| CRC | Communications Research Center |
| CW | Continuous Wave |
| dB | decibel |
| DREO | Defence Research Establishment Ottawa |
| EARS | Experimental Array Radar System |
| EMI | Electro-Magnetic Interference |
| IF | Intermediate Frequency |
| PIFS | Portable Interference Frequency Source |
| RF | Radio Frequency |
| SBC | Single Board Computer |
| SBS | Silo Beacon Source |
| VACP | Vertical Array Calibration Profiler |
| VDC | Volts Direct Current |